

U. S. NAVAL PERSONNEL RESEARCH ACTIVITY

SAN DIEGO, CALIFORNIA 92152

AD620026

TECHNICAL BULLETIN 65-15

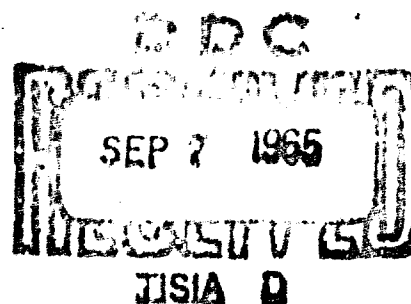
MAY 1965

A COMPUTER PROGRAM FOR THE MAXIMUM LIKELIHOOD ANALYSIS OF TYPES

John H. Wolfe

CLEARINGHOUSE FOR FEDERAL SCIENTIFIC AND TECHNICAL INFORMATION			
Hardcopy	Microfiche		
200-275-59 pp only			
ARCHIVE COPY			

DDC Ref Use only



NPRA



AN ACTIVITY OF THE BUREAU OF NAVAL PERSONNEL

**Best
Available
Copy**

A COMPUTER PROGRAM FOR THE
MAXIMUM LIKELIHOOD ANALYSIS OF TYPES

JOHN H. WOLFE

U.S. NAVAL PERSONNEL RESEARCH ACTIVITY
SAN DIEGO, CALIFORNIA 92152

BRIEF

This report contains a description of a computer program for estimating the parameters of a mixture of multivariate normal distributions with unknown frequencies, means, and covariances. The basic equations for the procedure are presented for the first time here, with their derivation omitted. An example with the results of the computer printout is described for an artificially constructed mixture of three bivariate normal distributions. The method of using the program and the Fortran listing are detailed in this report.

CONTENTS

BRIEF		PAGE
I	Identification	1
II	Purpose.	1
III	Restrictions	1
IV	Method	1
V	Usage.	10
VI	Example - Artificial Clusters.	12
	A. Listing of Input for Example.	17
	B. Computer Printout for Example	23
VII	Fortran Program Listing	30

A COMPUTER PROGRAM FOR THE MAXIMUM LIKELIHOOD ANALYSIS OF TYPES

I Identification

A. TYPE

- B. Written by John H. Wolfe, January 1963. Revised June, 1964.
- C. U.S. Naval Personnel Research Activity, San Diego, California.
- D. Coded entirely in FORTRAN II.

II Purpose

Given m scores on each of N individuals drawn from an unknown mixture of multivariate normal distributions, the program gives maximum-likelihood estimates of the means and covariances within each type, the relative frequency of each type, and the maximum likelihood of the sample for a given number of types.

III Restrictions

A. The program uses 14,554₈ words in the main program and 47,256₈ words in COMMON. One input magnetic tape (Unit #3), one output magnetic tape (Unit #2) and one tape for temporary binary storage (Unit #4) are required.

B. Restrictions on Parameters

Numbers of Variables ≤ 5

Number of Types ≤ 6

IV Method

The program solves the maximum-likelihood equations by one of four alternative iteration schemes, depending on a control card option.

Suppose that m measurements have been made on N individuals. Let X_{ik} be the i^{th} variable for individual k . Suppose the population from which the sample of N individuals is drawn is a mixture of r multivariate normal distributions. That is, the probability density, $f(\bar{x})$, is given by

$$f(\bar{x}) = \sum_{s=1}^r \lambda_s \alpha_s(\bar{x}), \text{ where } \sum_{s=1}^r \lambda_s = 1, \lambda_s \geq 0, \text{ and}$$

$$\alpha_s(\bar{x}) = \left(\frac{|\sigma_s^{-1}|}{(2\pi)^m} \right)^{1/2} e^{-1/2 \sum_{ij} (x_i - M_i^s) \sigma_s^{ij} (x_j - M_j^s)}$$

Here λ_s is the relative proportion of type s in population, $|\sigma_s^{-1}|$ is the determinant of the inverse of the covariance matrix for type s , and M_i^s is the mean of the i^{th} variable for type s .

Let us define $g_s(x) = \alpha_s(x) / f(x)$. SUBROUTINE DENSITY calculates α_s , f , and g_s for each individual. Define the "generalized sample moments" $\{\mu_{ijab}^{ps}\}$ as follows:

$$\mu_{ijab}^{ps} = \frac{1}{N} \sum_{k=1}^N x_{ik} x_{jk} x_{ak} x_{bk} g_s(x_k) g_p(x_k)$$

If a subscript is omitted or set to 0, the corresponding term on the right hand side of the equation is omitted. For example

$$\mu_{ijao}^{po} = \mu_{ija}^p = \frac{1}{N} \sum_{k=1}^N x_{ik} x_{jk} x_{ak} g_p(x_k)$$

$$\text{and } \mu_o^o = \frac{1}{N} \sum_{k=1}^N 1 = 1.$$

SUBROUTINE MOMENT computes a table of $\{\mu_{ijab}^{ps}\}$ with $b \leq a \leq j \leq i$ and $p \leq s$.

FUNCTION U (II, IJ, IA, IB, IP, IS) looks up the correct μ_{ijab}^{ps} from the two-dimensional table created by MOMENT, even when the inequalities on the indices are not satisfied.

The maximum likelihood estimates of the parameters are those values which maximize the function

$$L = \sum_{k=1}^N \log f(x_k) - \omega \left(\sum_{s=1}^r \lambda_s - 1 \right).$$

Setting the partial derivatives of the likelihood to zero results in the following equations:

$$f_o^s = \mu_o^s - 1 = 0$$

$$f_{oi}^s = \mu_i^s - \mu_o^s M_i^s = 0$$

$$f_{ij}^s = \mu_{ij}^s - M_i^s \mu_j^s - M_j^s \mu_i^s - \mu_o^s (\sigma_{ij}^s - M_i^s M_j^s) = 0$$

The moments $\{\mu_{ij}^s\}$ can also be differentiated as follows:

$$\frac{\partial \mu_{ij}^s}{\partial \lambda_p} = -\mu_{ij}^{ps}$$

$$\frac{\partial \mu_{ij}^s}{\partial M_a^p} = -\lambda_p \sum_{b=1}^m \sigma_p^{ab} (\mu_{ijb}^{ps} - M_b^p \mu_{ij}^{ps}) + \delta_{ps} \sum_{b=1}^m \sigma_p^{ab} (\mu_{ijb}^p - M_b^p \mu_{ij}^p)$$

$$\frac{\partial \mu_{ij}^s}{\partial \sigma_p^{ab}} = (1 - \frac{\delta_{ab}}{2}) \{-\lambda_p [\mu_{ij}^{ps} (\sigma_{ab}^p - M_a^p M_b^p) + M_a^p \mu_{ijb}^{ps} + M_b^p \mu_{ija}^{ps} - \mu_{ijab}^{ps}]$$

$$+ \delta_{ps} [\mu_{ij}^p (\sigma_{ab}^p - M_a^p M_b^p) + M_a^p \mu_{ijb}^p + M_b^p \mu_{ija}^p - \mu_{ijab}^p]\}$$

Where $\delta_{ps} = 1$ if $p=s$
0 otherwise

The derivatives of the moments are computed by

$$\text{FUNCTION DERV (II, IJ, IS, IA, IB, IP)} = \frac{\partial \mu_{ij}}{\partial \theta_{ab}^p},$$

Where $\theta_{oa}^p = M_a^p$, $\theta_{oo}^p = \lambda_p$, and $\theta_{ab}^p = \sigma_p^{ab}$ for $a \neq b$.

DERV calls on three functions, ONE, EM, SIG defined as follows:

$$\text{ONE (IA, IB)} = \delta_{ab}$$

$$\text{EM (II, IJ, IA, IS, IP)} = \sum_{b=1}^m \sigma_p^{ab} (\mu_{ijb}^{ps} - M_b^p \mu_{ij}^{ps})$$

$$\begin{aligned} \text{SIG} (IJ, IA, IB, IS, IP) = & \mu_{ij}^{ps} (\sigma_{ab}^p - M_a^p M_b^p) + M_a^p \mu_{ijb}^{ps} \\ & + M_b^p \mu_{ija}^{ps} - \mu_{ijab}^{ps} \end{aligned}$$

The function EM used with $s=0$ gives the second term in $\frac{\partial \mu_{ij}^s}{\partial M_a^p}$

and SIG used with $s=0$ gives the second term in $\frac{\partial \mu_{ij}^s}{\partial \sigma_p^{ab}}$.

The maximum likelihood equations $\{f_{ij}^s = 0\}$ can be solved in several ways. One iterative scheme is Newton-Raphson iteration, which solves the linear equations

$$\sum_{p=1}^r \sum_{b=1}^m \sum_{a=b}^m \frac{\partial f_{ij}^s}{\partial \theta_{ab}^p} \Delta \theta_{ab}^p = -f_{ij}^s$$

for $\Delta \theta_{ab}^p$.

On the next iteration $\theta_{ab}^{p'} = \theta_{ab}^p + \Delta \theta_{ab}^p$.

SUBROUTINE NEWTON computes the vector

$$B(IAT) = -f_{ij}^s \text{ and the matrix of coefficients}$$

$$A(IAT, JAT) = \partial f_{ij}^s / \partial \theta_{ab}^p.$$

SUBROUTINE MATINV, a standard SHARE routine, solves the linear equations for $\Delta \theta_{ab}^p$ and stores the result in the vector B (IAT).

SUBROUTINE RAPISON computes the values of the parameters for the next iteration. First it determines if any of the increments are so large that $\lambda_p + \Delta \lambda_p < 0$ or > 1 . If so, the increment vector B is shortened until no $\Delta \lambda_p$ moves λ_p more than half the interval between λ_p and a boundary point.

That is, if $\Delta \lambda_p < 0$, $\Delta \lambda_p > -\frac{\lambda_p}{2}$, and if $\Delta \lambda_p > 0$, $\Delta \lambda_p < \frac{1-\lambda_p}{2}$. After the $\Delta \theta_{ab}^p$

are shortened, they are added to the old estimates of the parameters to obtain new estimates for the next iterations.

. The main routine also plays a role in shortening the increment vector. If the new likelihood is less than the previous one, or if the determinant of one of the covariance matrices as determined by RAPHISON is negative, then the increment vector B is shortened to half its previous value and subroutine RAPHISON is entered again.

Several alternative versions of $\{f_{ij}^S\}$ can be written.

At the maximum-likelihood points

$$f_{00}^S = \mu_0^S - 1 = 0, \text{ hence } \mu_0^S = 1.$$

$$f_{0i}^S = \mu_i^S - \mu_0^S M_i^S = \mu_i^S - M_i^S = 0, \text{ hence } \mu_i^S = M_i^S.$$

Substituting for μ_i^S and μ_0^S in f_{ij}^S , we have

$$f_{ij}^S = \mu_{ij}^S - M_i^S M_j^S.$$

When METH = 1 on the control card, the subroutine NEWTON iteratively solves the equations

$$1f_{00}^S = \mu_0^S - 1 = 0.$$

$$1f_{0i}^S = \mu_i^S - M_i^S = 0.$$

$$1f_{ij}^S = \mu_{ij}^S - M_i^S M_j^S = 0.$$

When METH = 2, subroutine NEWTON solves the original set of equations, hereafter referred to as $\{2f_{ij}^S = 0\}$

When METH = 3, the equations used are

$$3f_{00}^S = 1/\mu_0^S (2f_{00}^S) = 1 - 1/\mu_0^S = 0.$$

$$3f_{0i}^S = 1/\mu_0^S (2f_{0i}^S) = \mu_i^S/\mu_0^S - M_i^S = 0.$$

$$3f_{ij}^S = 1/\mu_0^S (2f_{ij}^S) = \mu_{ij}^S - M_i^S \mu_j^S/\mu_0^S - M_j^S \mu_i^S/\mu_0^S - (c_{ij}^S - M_i^S M_j^S) = 0.$$

All three types of equations have the same solutions but their radii of convergence may differ.

For METH = 2,

the function $\{2f_{ij}^s\}$ can be summarized in the following formula:

$$2f_{ij}^s = \mu_{1j}^s - \sigma_{io} - (1 - \delta_{io})\{\theta_i^s \mu_j^s + (1 - \delta_{jo})[\mu_o^s(\sigma_{ij}^s - \theta_i^s \theta_j^s) + \theta_j^s \mu_i^s]\}.$$

The partial derivatives of f_{ij}^s are then easily written as

$$\begin{aligned} \frac{\partial f_{ij}^s}{\partial \theta_{ab}^p} &= \frac{\partial \mu_{ij}^s}{\partial \theta_{ab}^p} - (1 - \delta_{io})\{\theta_j^s \frac{\partial \mu_j^s}{\partial \theta_{ab}^p} + (1 - \delta_{jo})[\frac{\partial \mu_o^s}{\partial \theta_{ab}^p}(\sigma_{ij}^s - \theta_i^s \theta_j^s) + \theta_j^s \frac{\partial \mu_i^s}{\partial \theta_{ab}^p}]\} \\ &- \delta_{ps}(1 - \delta_{io})\{\delta_{bo}[\delta_{ia} \mu_j^s + (1 - \delta_{jo})(\delta_{ja} \mu_i^s - \delta_{ia} \mu_o^s \theta_j^s - \delta_{ja} \mu_o^s \theta_i^s)] \\ &- (1 - \frac{\delta_{ab}}{2})(1 - \delta_{bo})(1 - \delta_{jo}) \mu_o^s (\sigma_{ia}^p \sigma_{jb}^p + \sigma_{ib}^p \sigma_{ja}^p)\} \end{aligned}$$

where $0 \leq j \leq m$ and $0 \leq b \leq m$.

The above formulas are used in METH = 2. If METH = 1, the functions

$$\begin{aligned} 1f_{oo}^s &= \mu_o^s - 1 \\ 1f_{oi}^s &= \mu_i^s - M_i^s \\ 1f_{ij}^s &= \mu_{ij}^s - (\sigma_{ij}^s + M_i^s M_j^s) \end{aligned}$$

are summarized by the formulas

$$\begin{aligned} 1f_{ij}^s &= \mu_{ij}^s - \delta_{io} - (1 - \delta_{io})(\delta_{jo} \theta_i^s + (1 - \delta_{jo})(\tau_{ij}^s + \theta_i^s \theta_j^s)), \text{ and} \\ \frac{\partial 1f_{ij}^s}{\partial \theta_{ab}^p} &= \frac{\partial \mu_{ij}^s}{\partial \theta_{ab}^p} - (1 - \delta_{io})\delta_{ps}\{\delta_{bo} \delta_{jo} \delta_{ia} + (1 - \delta_{jo})(\delta_{ia} \theta_j^s + \delta_{ja} \theta_i^s)\} \\ &- (1 - \delta_{ab}/2)(1 - \delta_{bo})[1 - \delta_{jo}](\sigma_{ia}^p \sigma_{jb}^p + \tau_{ib}^p \sigma_{ja}^p). \end{aligned}$$

The functions $\{f_{ij}^s\} = \{f_{ij}^s/\mu_0^s\}$.

$$\text{Hence } \frac{\partial f_{ij}^s}{\partial \mu_{ab}^p} = \frac{1}{\mu_0^s} \frac{\partial f_{ij}^s}{\partial \mu_{ab}^p} - \frac{f_{ij}^s}{(\mu_0^s)^2} \frac{\partial \mu_0^s}{\partial \mu_{ab}^p}.$$

The maximum likelihood Newton-Raphson iteration equations give:

$$\sum_{abp} \frac{\partial^2 f_{ij}^s}{\partial \mu_{ab}^p} \Delta \mu_{ab}^p = - f_{ij}^s.$$

Multiplying by μ_0^s and substituting, we have

$$\sum_{abp} \left[\frac{\partial^2 f_{ij}^s}{\partial \mu_{ab}^p} - f_{ij}^s \left(\frac{1}{\mu_0^s} \frac{\partial \mu_0^s}{\partial \mu_{ab}^p} \right) \right] \Delta \mu_{ab}^p = - f_{ij}^s.$$

Thus the equations for METH = 3 are readily calculated from the equations for METH = 2. Only the matrix of coefficients has to be

changed, simply by subtracting $f_{ij}^s \left(\frac{1}{\mu_0^s} \frac{\partial \mu_0^s}{\partial \mu_{ab}^p} \right)$.

Instead of Newton-Raphson iteration, a method of successive substitutions may be used for finding M_i^s and σ_{ij}^s .

Since $f_{oi}^s = 0 = \mu_i^s - \mu_0^s M_i^s$, the value of M_i^s for the next iteration is defined as $M_i^s = \mu_i^s / \mu_0^s$.

Since $f_{ij}^s = 0 = \mu_{ij}^s - M_i^s \mu_1^s - M_j^s \mu_1^s - \mu_0^s (\sigma_{ij}^s - M_i^s M_j^s)$ we can solve for σ_{ij}^s after first substituting $M_i^s \mu_0^s$ for μ_1^s :

$$\sigma_{ij}^s = \mu_{ij}^s / \mu_0^s - M_i^s M_j^s, \text{ where } M_i^s \text{ are the new values } = \mu_i^s / \mu_0^s.$$

The new values of λ_p can be determined by Newton-Raphson iteration using only the equations $\{f_{00}^s = \mu_0^s - 1 = 0\}$

Differentiation of these equations leads to the system

$$\sum_{p=1}^r \frac{\partial \mu_0^s}{\partial \lambda_p} \Delta \lambda_p = - \mu_0^s + 1$$

$$\text{or } \sum_{p=1}^r u_{ps} \Delta \lambda_p = u_0^s - 1$$

When the control card METH = 0, subroutine NEWTON determines the increments associated with successive substitutions.

Experience with the program seems to indicate that the NEWTON-Raphson iteration schemes have very small radii of convergence as compared with the successive substitution methods. Theoretically, however, the Newton-Raphson iteration should converge more rapidly once within its radius. Therefore, provision has been made in the program for running a fixed number of iterations with successive substitutions so as to get improved initial estimates and then switching to Newton-Raphson methods for the remaining iterations. If the control card option is -1, -2, or -3, then a certain number of iterations by successive substitution will be used before Newton-Raphson iteration by methods 1, 2, 3 respectively. The control card number IDUMP specifies the number of preliminary iterations.

The subroutine INITIAL determines the initial values of parameters preliminary to iteration. The proper determination of initial values is crucial to the successful convergence of any iteration method. The initial values determined by INITIAL are quite crude, and the researcher may wish to write his own version of INITIAL after some experimentation. The present version also allows the user to specify his own guesses of the initial values of the parameters of certain types.

The subroutine INITIAL takes a sample of 100 individuals and subjects them to a crude clustering procedure. For each individual, a count is made of the number of other individuals within a "box" two standard deviations on a side around it. That is, for individual k , the number of individuals j is counted such that $|x_{ik} - x_{ij}| \leq \sigma_i$ for all $i = 1, 2, \dots, m$. The individual with the highest count is made the centroid of the first cluster--that is, his scores are the initial estimates of the means for the first type.

The individuals in the first cluster are erased from the sample and the procedure is repeated with the remaining individuals.

The initial estimates of the λ_s are all equal to $1/r$, where r is the number of types.

The initial estimates of the covariances are the same for each type and are equal to the covariances computed from the sample of 100 taken as a whole.

After each iteration the subroutine RESULT prints the current estimates of the parameter for each type. At the end of the last iteration, the program PLACE gives the probabilities of membership in each type for each individual.

These probabilities are:

$$P(\text{individual } k \text{ \& Type } S) = \lambda_s g_s(x_k).$$

A few words should be said about the indexing used within the program. First of all, the indices of the moments do not range from 0 to m and 0 to r as in our equations, but from 1 to m+1 and 1 to r+1. Thus the value of $U(2, 3, 1, 1, 1, 4) = \mu_{12}^3$. The moments are conveniently calculated by

$$\text{setting } Z_{i+1} = x_i \text{ for each } x_{ik}$$

$$\text{and } Z_1 = 1.0. \text{ Similarly } G(1) = 1.0$$

$$\text{and } G(2) \text{ is the relative density for type 1, } g_1(x_k).$$

$$\text{The values for PERS } (K) = \lambda_{k-1}$$

$$\text{COV}(I, J, K) = \sigma_{ij}^{k-1} \text{ and } \text{COVIN}(I, J, K) = \sigma_{k-1}^{ij}.$$

$$\text{also } \text{AV}(I, K) = M_i^{k-1}.$$

The routine MOMENT collapses the (μ_{ijab}^{ps}) into a two dimensional array.

The single index KL is uniquely related to p and s and the single index IJ'N is uniquely related to the indices i, j, a, and b.

In general, suppose we have an array indexed as follows:

There are M indices. The first index varies from 1 to N. Each succeeding index varies from 1 to the preceding index.

$$1 \geq IX(1) \geq \dots \geq IX(M)$$

Let $S(N, M)$ = number of elements in this array.

$$\text{Then } S(N, M) = \sum_{I=1}^N S(I, M-1)$$

$$\text{and } S(N, M) = \frac{(N+M-1)!}{M!(N-1)!} = \binom{N+M-1}{M}.$$

Let $IX(1) \geq IX(2) \geq \dots \geq IX(M)$ be a sequence of indices for a particular element of the array. The one-dimensional index of the element is

$$K = 1 + \sum_{I=1}^M \sum_{J=1}^I (IX(M-I+1) - 2 + J) / J$$

$$\text{or } K = 1 + \sum_{I=1}^M S(IX(M-I+1) - 1, I).$$

These formulas are used by the Function U to look up values of μ_{ijab}^{ps} .

V Usage

- A. Input (TAPE Unit 3, BCD-card images)
1. Title Card in columns 1-72, any alphanumeric characters.
 2. Control Card

COLS:	NAME	DEFINITION
1-4	MX	Number of variables
5-12	NX	Sample size
13-16	IRM	Number of Types Assumed. If IRM = 0, 6 analyses will be done assuming 1, 2, 3, 4, 5, 6 types.
17-20	ITERM	Maximum Number of iterations. If blank, ITERM is set to 50.
21-28	CONV	Criterion of Convergence which all parameters must satisfy between successive iterations. If blank, CONV is set to .0001.
29-32	IRUN	=1 if every iteration is printed, = 0 if only the last iteration is printed
33-36	MFTI	=0 if successive substitutions is used <u>+1</u> , <u>+2</u> , <u>+3</u> , if various Newton-Raphson methods are used.
37-40	IDUMP	The number of preliminary iterations by successive substitutions before Newton-Raphson iteration for MFTI = -1, -2, or -3.

3. Variable Format Card. This is an ordinary FORTRAN Variable Format Card according to which the data will be read.

4. Data Deck
N sets of one or more cards per individual.

5. Initial Estimates of Parameters (optional)

- (a) Estimate control card
 - Cols 1-4 = K = TYPE # (1 through 6)
 - Cols 5-8 = λ_k = Proportion of population of type K (all 4 digits assumed after decimal point)
- (b) Means for type K
8 digits per mean, last 4 digits assumed after decimal point.
- (c) Standard deviations for type K (same format as (b))
- (d) Correlation matrix for type K (1 row per card, same format as (b))
- (e) Estimate control card for another type, etc.

A blank card terminates the reading of initial estimates. If no initial estimates are to be read, one blank card must be read. The sets of initial estimates do not have to be present for all types. For example, a set of initial estimates for type 5 may be followed by a set for type 3 followed by a blank card. If any initial parameters are read

for a given type, all parameters must be read for that type. For example, initial estimates of means for type 3 must be followed by initial estimates of standard deviations and correlations for type 3.

If no initial estimates of parameters are read in, the computer will generate its own by a clustering procedure.

Multiple runs may be made at one time by placing one batch of input cards [A(1) to A(5)] followed by another batch. The last batch of data must be followed by two blank cards.

B. Recommended typical usage.

Under most conditions, the control card A-2 will be blank in all but the first 12 columns. The composition of the input will be:

Title Card
Control Card (Cols 1-12 only)
Variable Format Card
Data Cards
3 blank cards.

Another highly successful set of control parameters is METH = -1 or -2 with IDUMP = 40

C. Output (Tape No 2)

The natural logarithm of the likelihood is printed for each

iteration: $\sum_{k=1}^N \log f(x_k) - N \left(\sum_{s=1}^r \lambda_s - 1 \right)$. If the likelihood on an

iteration is not greater than that on all previous iterations, the computer prints out "Iteration--diverges". The parameters for each type are printed out on either the last iteration or on every iteration, depending on the value of IRUN. At the end of the last iteration, the probabilities of type membership for each individual, $\lambda_{sg}(x_k)$, are printed. If sense switch 3 is down, the computer dumps out the moments $\{\mu_{ijab}^{ps}\}$ and the matrices $\{\sigma_{ij}^s\}$ and $\{\sigma_{ij}^{ij}\}$ following subroutine MOMENT, the matrix A of coefficients of the Newton-Raphson iteration and the vector $\{-f_{ij}^s\}$ following subroutine NEWTON, and the matrices A^{-1} and the vector $\{\delta \theta_{ah}^p\}$ following the subroutine MATINV.

D. Time estimates.

For METH = 0, time per iteration in seconds is

$$T = .14 N (m+1)(m+2)(r+1)$$

For METH = 0, time per iteration in seconds is

$$T = .0075 (m+1)(m+2)(m+3)(m+4)(r+1)(r+2)N$$

$$+ .00002[(m+1)(m+2)r]^3 N$$

A typical run with N = 225, r = 1, 2, 3, 4, 5, 6, m = 2 required 50 minutes by METH = 0 on the CDC 1604.

E. Suggestions for reducing running time.

1. By intuition, cluster analysis of variables, factor analysis, or any other means, reduce the number of variables to those that are really important for discriminating types.

2. Use any good prior information from theoretical hypotheses or any other classification-clustering procedures to provide initial estimates.

3. If you have a large sample, say 10,000, don't run it all at once. Take a small sample, say 100 to 200 cases at random and run an analysis using a weak convergence criterion, say .01. Get an idea of how many types there are and some initial estimates from the small sample. Run the entire set of data using these initial estimates on specified numbers of types. The hypothesis H_r that there are r types can be tested against the alternative, H_{r-1} that there are $r-1$ types by the χ^2 test

$$\chi^2_{d.f.} = -2 \log \lambda = 2(L_r - L_{r-1})$$

$$\text{with d.f.} = (m+1)(m+2)/2$$

where L_r = likelihood printed for r types. Thus the

χ^2 test of the likelihood ratio on the small sample may give a good guess as to how many types there are. A final test on the entire sample could be performed specifying r , $r+1$, and $r-1$ types on the runs, where r is the hypothesized number of types.

VI Example - Artificial Clusters

To test this program, an example was constructed consisting of three artificial clusters in two dimensions. The points in each cluster were generated by a pseudo-random normal deviate generator. The characteristics of each cluster are given in Table 1 below. The results of the computer printout are summarized in Table 2. The points are plotted in Figure 1. The 75 points in Cluster 1 are designated by triangles in Figure 1, the 50 points in Cluster 2 are designated by squares, and the 100 points in Cluster 3 are designated by circles. Drawn around some of the squares, circles, and triangles are larger squares, circles and triangles. The larger symbols give the classification assigned by the computer program with 3 specified types. If a point does not have two symbols around it, it was correctly classified by the computer. It is evident from the figure that most points were correctly classified, and the computer's types have clear cut boundaries whereas the actual clusters overlap to some degree.

TABLE 1
Characteristics of Artificial Clusters

	Cluster 1		Cluster 2		Cluster 3	
Number/225	.333		.222		.444	
MEANS	.05	-1.33	-.83	1.64	1.41	.85
S.D. 's	1.26	.49	.87	1.04	.92	.97
r_{12}	.1590		.4040		.4743	

TABLE 2
Characteristics of Types from Computer Program

	Type 2		Type 3		Type 1	
λ_p	.346		.170		.484	
Means	.20	-1.31	-1.11	1.79	1.19	.93
S.D. 's	1.28	.50	.83	1.12	1.04	.91
r_{12}	.2462		.7168		.5231	

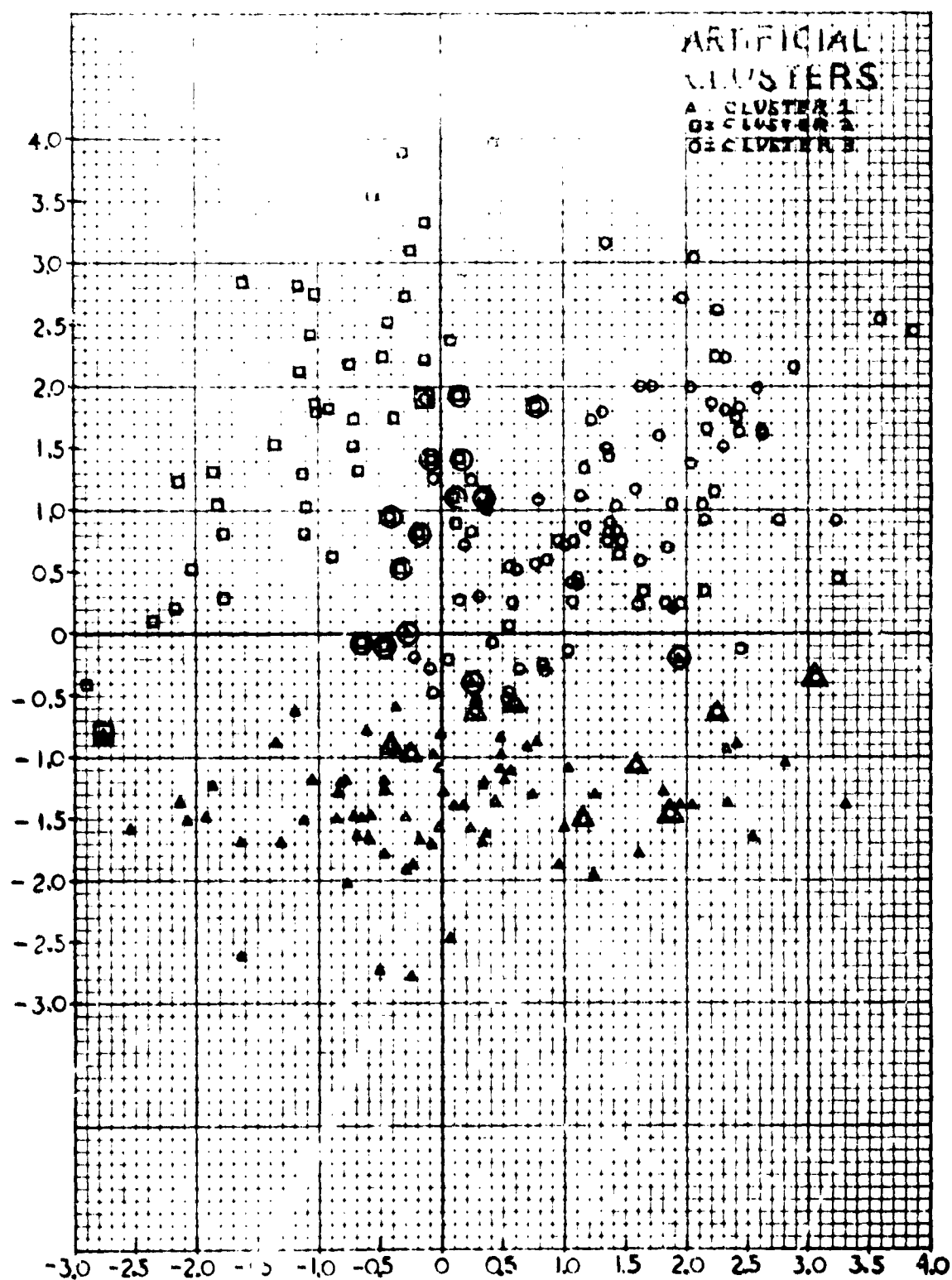


TABLE 3
Likelihoods and χ^2 for Numbers of Types

Number of Types	Natural Logarithm of Likelihood	χ^2 (with 6d.f.) $=2(L_R - L_{R-1})$	P
1	-380.48930		
2	-358.96468	43.04924	.114 X 10 ⁻⁶
3	-340.36400	37.20136	.161 X 10 ⁻⁵
4	-334.83078	11.06644	.863 X 10 ⁻¹
5	-325.63847	18.38462	.534 X 10 ⁻²
6	-318.02872	15.21950	.186 X 10 ⁻¹

The data cards for input are listed in Section B and the output is given in Section C.

A previous run with METH = 0, IRM = 0 (not given here) gave the likelihoods for 1 to 6 types. These are presented in Table 3, along with associated χ^2 values. The results indicate that the hypothesis that there are only two types can be rejected against the hypothesis that there are three types; but the hypothesis that there are three types cannot be rejected against the alternative that there are four types.

In order to save space in this report, only a run with IRM = 3 is given in Sections B and C.

The method used was METH = -1, which has 10 preliminary iterations by successive substitutions followed by Newton-Raphson iteration. The previous unpublished run with METH = 0 took 45 iterations to converge, while the one presented here took only 17. The difference can be attributed to the superior convergence rate of Newton-Raphson methods. However, other computer runs with IRM = 0, METH = -1 and IDUMP = 10, 30, or 40 sometimes failed to converge at all, once the Newton-Raphson procedure was started. If the likelihoods have not converged to 0.1 by the successive substitutions, then Newton-Raphson iteration often fails. Thus the initial estimates must be quite accurate if Newton-Raphson iteration is to work. So far the various methods -1, -2, -3, appear to work equally well. All three converged in exactly 17 iterations in the present example. The run reported here took 13 minutes on the CDC 1604.

SECTION VI

A. Listing of Input for Example

ARTIFICIAL CLUSTERS METHOD- 1 PRELIMINARY ITERATIONS

2	265	3	-1	-6	
(264.2)					001
-058-141					002
-068-167					003
103-317					004
-057-002					005
-012-151					006
-016-165					007
014-157					008
-070-165					009
-110-150					010
010-075					011
-214-131					012
014-243					013
211-100					014
272-090					015
-047-173					016
-042-118					017
-241-157					018
211-134					019
240-080					020
311-138					021
011-125					022
-043-129					023
-217-140					024
113-108					025
143-122					026
014-114					027
015-160					028
-164-146					029
-044-148					030
042-088					031
-077-200					032
108-134					033
-116-085					034
-130-168					035
121-191					036
-048-113					037
112-129					038
243-161					039
-036-095					040
-080-110					041
-040-162					042
-003-104					043
-072-141					044
-276-077					045
049-107					046
080-119					047
011-139					

-182-257
 -186-121
 34-116
 -184-186
 24-272
 77-177
 43-144
 41-144
 49-121
 -108-115
 78-286
 61-273
 60-270
 16-135
 232-143
 66-146
 -121-260
 24-186
 32-169
 49-233
 25-234
 54-127
 48-280
 94-185
 100-152
 161-174
 7-296
 26-182
 70 134
 47 229
 32 278
 77 186
 73 177
 -107 245
 -235-212
 -219 225
 59 357
 -284 167
 -179 281
 -111 125
 13 145
 74 220
 71 155
 46 261
 68-204
 14 337
 -115 292
 -114 215
 -179 331
 25 312

48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97

41 698
 35 055
 -103 280
 -292-038
 92 184
 -205 053
 -186 108
 18 084
 -215 128
 45 400
 49-000
 -161 288
 7 143
 -111 082
 11 111
 90 066
 34 112
 -113 133
 13 196
 11 091
 -186 130
 106 240
 -103 187
 40 178
 -138 157
 11 224
 32 391
 -101 185
 63-027
 131 180
 62 052
 275 092
 107 042
 208 306
 38 105
 218 137
 113-150
 139 147
 27 110
 14 190
 60-054
 240 174
 161 200
 141 084
 86 061
 7 123
 134 151
 172 200
 230 153
 160 025

98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147

13 030
 17 071
 214 107
 135 078
 224-063
 204 224
 103-012
 9-049
 56 056
 181 025
 163 060
 189 106
 261 163
 205 139
 193 126
 359 255
 118 187
 109 074
 2.1 038
 23 083
 322 093
 11 109
 54-042
 111 117
 10-028
 112 040
 106 029
 211 093
 148 118
 23 128
 6 027
 136 083
 79 059
 2-020
 31-089
 139 092
 287 217
 44-002
 241 185
 179 160
 96 079
 58 009
 221 188
 232 180
 222 117
 83-027
 148 078
 133 116
 197 270
 25-095

148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197

203 200
80 110
175 122
213 225
245 167
261 161
23-018
189 022
218 169
226 261
96 018
259 197
31 032
112 141
385 246
306-035
242-010
142 106
158-102
107 076
24-054
144 064
324 046
82-022
163 037
91-050
182 071
175-143

198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225

BLANKOC'
BLANKOC
BLANKOO.

SECTION VI

B. Computer Printout for Example

ITERATION 1, LIKELIHOOD OF	3 TYPES IN THIS SAMPLE	•	0.42371292E+03
ITERATION 2, LIKELIHOOD OF	3 TYPES IN THIS SAMPLE	•	0.37105540E+03
ITERATION 3, LIKELIHOOD OF	3 TYPES IN THIS SAMPLE	•	0.36744158E+03
ITERATION 4, LIKELIHOOD OF	3 TYPES IN THIS SAMPLE	•	0.36440563E+03
ITERATION 5, LIKELIHOOD OF	3 TYPES IN THIS SAMPLE	•	0.36094806E+03
ITERATION 6, LIKELIHOOD OF	3 TYPES IN THIS SAMPLE	•	0.35693192E+03
ITERATION 7, LIKELIHOOD OF	3 TYPES IN THIS SAMPLE	•	0.35270142E+03
ITERATION 8, LIKELIHOOD OF	3 TYPES IN THIS SAMPLE	•	0.34901014E+03
ITERATION 9, LIKELIHOOD OF	3 TYPES IN THIS SAMPLE	•	0.34638251E+03
ITERATION 10, LIKELIHOOD OF	3 TYPES IN THIS SAMPLE	•	0.34476925E+03
ITERATION 11, LIKELIHOOD OF	3 TYPES IN THIS SAMPLE	•	0.34381568E+03
ITERATION 12, LIKELIHOOD OF	3 TYPES IN THIS SAMPLE	•	0.34657528E+03
ITERATION 11 DIVERGES			
ITERATION 13, LIKELIHOOD OF	3 TYPES IN THIS SAMPLE	•	0.34173161E+03
ITERATION 14, LIKELIHOOD OF	3 TYPES IN THIS SAMPLE	•	0.34044823E+03
ITERATION 15, LIKELIHOOD OF	3 TYPES IN THIS SAMPLE	•	0.34037172E+03
ITERATION 16, LIKELIHOOD OF	3 TYPES IN THIS SAMPLE	•	0.34036401E+03
ITERATION 17, LIKELIHOOD OF	3 TYPES IN THIS SAMPLE	•	0.34036400E+03

MAXIMUM-LIKELIHOOD ANALYSIS OF TYPES
ARTIFICIAL CLUSTERS METHOD-1 1 PRELIMINARY ITERATIONS

SAMPLE SIZE = 220
NUMBER OF VARIABLES = 2
NUMBER OF TYPES = 3

ITERATION NUMBER 17

LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = $7.34036400E+03$

CHARACTERISTICS OF THE WHOLE SAMPLE

	MEANS
.46	.50
	STANDARD DEVIATIONS
1.38	1.47
	CORRELATIONS
1.0000	.1849
.1849	1.0000

CHARACTERISTICS OF TYPE 1

THE PROPORTION OF THE POPULATION FROM THIS TYPE = .484

	MEANS
1.19	.93
	STANDARD DEVIATIONS
1.14	.91

		CORRELATIONS
1.0000	.5231	
.5231	1.0000	

CHARACTERISTICS OF TYPE 2

THE PROPORTION OF THE POPULATION FROM THIS TYPE = .346

		MEANS
.20	-1.31	
		STANDARD DEVIATIONS
1.28	.90	
		CORRELATIONS
1.0000	.2462	
.2462	1.0000	

CHARACTERISTICS OF TYPE 3

THE PROPORTION OF THE POPULATION FROM THIS TYPE = .170

		MEANS
-1.11	1.79	
		STANDARD DEVIATIONS
.83	1.12	
		CORRELATIONS
1.0000	.7160	
.7160	1.0000	

PROBABILITIES OF TYPE MEMBERSHIP

1	.41	.957	.32
2	.21	.972	.32
3	.730	.981	.32
4	.457	.981	.32
5	.20	.971	.32
6	.21	.971	.32
7	.35	.974	.32
8	.25	.975	.32
9	.31	.975	.32
10	.223	.975	.32
11	.23	.965	.32
12	.19	.991	.32
13	.484	.984	.32
14	.175	.970	.32
15	.21	.970	.32
16	.172	.975	.32
17	.015	.984	.37
18	.184	.984	.32
19	.23	.977	.32
20	.15	.971	.32
21	.157	.971	.32
22	.154	.945	.32
23	.14	.971	.32
24	.154	.944	.32
25	.153	.987	.32
26	.17	.920	.32
27	.125	.98	.32
28	.121	.971	.32
29	.134	.945	.32
30	.134	.942	.32
31	.114	.984	.32
32	.107	.991	.32
33	.155	.914	.35
34	.121	.974	.32
35	.15	.995	.32
36	.187	.913	.32
37	.122	.975	.32
38	.101	.999	.32
39	.494	.954	.32
40	.175	.93	.32
41	.124	.974	.32
42	.111	.984	.32
43	.041	.959	.32
44	.114	.944	.32
45	.184	.914	.32
46	.164	.935	.32
47	.137	.961	.32
48	.118	.985	.32
49	.130	.942	.32
50	.167	.933	.32
51	.119	.98	.32
52	.113	.987	.32
53	.142	.954	.32
54	.128	.972	.32
55	.137	.961	.32
56	.169	.931	.32
57	.175	.927	.32
58	.151	.947	.32
59	.195	.977	.32
60	.114	.984	.32

61	.041	.959	.000
62	.004	.996	.000
63	.034	.964	.000
64	.342	.626	.031
65	.017	.983	.000
66	.014	.984	.000
67	.137	.863	.000
68	.744	.254	.000
69	.081	.919	.000
70	.217	.783	.000
71	.007	.993	.000
72	.014	.986	.000
73	.004	.996	.000
74	.144	.855	.000
75	.014	.984	.000
76	.184	.000	.014
77	.024	.000	.986
78	.004	.000	.994
79	.992	.000	.008
80	.042	.000	.958
81	.001	.000	.999
82	.023	.016	.961
83	.017	.021	.982
84	.000	.000	1.000
85	.000	.000	1.000
86	.015	.000	.985
87	.000	.000	.911
88	.927	.000	.073
89	.000	.000	.991
90	.094	.000	.906
91	.004	.000	.994
92	.904	.066	.028
93	.001	.000	.999
94	.000	.000	1.000
95	.003	.000	.997
96	.054	.001	.944
97	.002	.000	.998
98	.784	.000	.212
99	.963	.001	.036
100	.000	.000	1.000
101	.000	.051	.941
102	.014	.000	.984
103	.014	.000	.986
104	.004	.000	.994
105	.960	.000	.040
106	.002	.000	.998
107	.001	.000	.999
108	.910	.073	.028
109	.000	.000	1.000
110	.810	.000	.190
111	.160	.000	.831
112	.970	.000	.021
113	.477	.000	.523
114	.995	.000	.005
115	.034	.000	.964
116	.594	.000	.406
117	.991	.000	.009
118	.004	.000	.996
119	.984	.000	.016
120	.000	.000	.991
121	.160	.000	.840
122	.007	.000	.993
123	.104	.000	.894
124	.000	.000	1.000
125	.011	.000	.989
126	.000	.200	.800

127	1.000	.000	.000
128	.990	.001	.000
129	1.000	.000	.000
130	.997	.003	.000
131	1.000	.000	.000
132	.997	.003	.003
133	1.000	.000	.000
134	.013	.987	.000
135	1.000	.000	.000
136	.993	.000	.007
137	.300	.000	.490
138	.477	.523	.000
139	1.000	.000	.000
140	1.000	.000	.000
141	1.000	.000	.000
142	.990	.001	.000
143	.981	.001	.117
144	1.000	.000	.000
145	1.000	.000	.000
146	1.000	.000	.000
147	.987	.013	.000
148	.994	.004	.001
149	.997	.003	.003
150	1.000	.000	.000
151	1.000	.000	.000
152	.105	.895	.000
153	1.000	.000	.000
154	.893	.107	.000
155	.575	.425	.000
156	.990	.001	.000
157	.984	.016	.000
158	.990	.001	.000
159	1.000	.000	.000
160	1.000	.000	.000
161	1.000	.000	.000
162	1.000	.000	.000
163	1.000	.000	.000
164	1.000	.000	.000
165	1.000	.000	.000
166	.992	.008	.000
167	.997	.003	.003
168	1.000	.000	.000
169	.981	.019	.019
170	.638	.362	.000
171	1.000	.000	.000
172	.805	.195	.000
173	.997	.003	.000
174	.993	.007	.000
175	1.000	.000	.000
176	1.000	.000	.000
177	.980	.000	.020
178	.993	.007	.000
179	1.000	.000	.000
180	.990	.001	.000
181	.868	.132	.000
182	.184	.816	.000
183	1.000	.000	.000
184	1.000	.000	.000
185	.951	.049	.000
186	1.000	.000	.000
187	1.000	.000	.000
188	1.000	.000	.000
189	.475	.525	.000
190	1.000	.000	.000
191	1.000	.000	.000
192	1.000	.000	.000

193	.786	.214	.000
194	1.000	.000	.000
195	.897	.000	.103
196	1.000	.000	.000
197	.151	.849	.000
198	1.000	.000	.000
199	1.000	.000	.000
200	1.000	.000	.000
201	1.000	.000	.000
202	1.000	.000	.000
203	1.000	.000	.000
204	.876	.123	.001
205	.979	.021	.000
206	1.000	.000	.000
207	1.000	.000	.000
208	.985	.015	.000
209	1.000	.000	.000
210	.995	.005	.000
211	1.000	.000	.000
212	1.000	.000	.000
213	.135	.865	.000
214	.709	.291	.000
215	1.000	.000	.000
216	.041	.959	.000
217	1.000	.000	.000
218	.506	.494	.000
219	.990	.001	.000
220	.982	.018	.000
221	.833	.167	.000
222	.995	.005	.000
223	.538	.462	.000
224	1.000	.000	.000
225	.007	.993	.000

Best Available Copy

VII Fortran Program Listing

```

PROGRAM TYPE TYPE0000
DIMENSION Y(5,7), PFR(7), PERS(7), AV(5,7), COV(5,5,7), COVIN(5,5,7), ALPHA(7), DETERM(7), G(7), IX(4), ID(4), V(126,20), A(126,126), B(126,126), NV(126), RECORD(12), SD(5), XSAM(5,10), MAT(10,10) TYPE0001
DIMENSION FMT(12) TYPE0002
COMMON X,7,PFR,PERS,AV,COV,COVIN,ALPHA,DETERM,G,V,A,H,NV,RECORD,SD TYPE0003
1,MX,MX1,IR,YIR,IR1,IJMN,KLX,NY,YNX,JATNX,CONV,ITERM,LX,XSAM,MAT,ET TYPE0004
2PH,IX,II,ITER,PROB,METH,JDUMP TYPE0005
EQUIVALENCE (XSAM,V),(A,MAT,Y) TYPE0006
1 READ INPUT TAPE 3,1,1,(RECORD(1),1=1,12),MX,NX,IRM,ITERM,CONV,IRUN TYPE0007
1,MOTH,JDUMP TYPE0008
101 FORMAT(12A4/14,10,214,F0.0,314) TYPE0009
IF(MX) 23,23,2 TYPE0010
C MX=NUMBER OF VARIABLES. TYPE0011
C THE PROGRAM READS BATCHES OF DATA UNTIL 2 BLANK CARDS ARE ENCOUNTERED. TYPE0012
C 1=NO. TYPE0013
2 IF(ITERM)3,3,4 TYPE0014
C ITERM=MAXIMUM NUMBER OF PERMISSIBLE ITERATIONS. TYPE0015
3 ITERM=5 TYPE0016
4 IF(CONV)5,5,6 TYPE0017
C CONVE=CRITERION OF CONVERGENCE WHICH ALL PARAMETERS MUST SATISFY BETWEEN SUCCESSIVE ITERATIONS. TYPE0018
5 CONVE=0.0001 TYPE0019
6 MX1=MX+1 TYPE0020
REWIND 4 TYPE0021
XNX=NX TYPE0022
NEV=X*INTF(YNX/100.)+1 TYPE0023
IJMN=(MX1+(MX1+1)+(MX1+2)+(MX1+3))/24 TYPE0024
KRMN=0 TYPE0025
LX=0 TYPE0026
READ INPUT TAPE 3,102,(FMT(1),1=1,12) TYPE0027
102 FORMAT(12A4) TYPE0028
7 CALL BLOCK(MX,3000,KRMN,KST,KEND,LONG) TYPE0029
DO 8 J=1,LONG TYPE0030
8 READ INPUT TAPE 3,FMT,(X(1,J),1=1,MX) TYPE0031
DO 9 J=1,LONG,NEV TYPE0032
LX=LX+1 TYPE0033
DO 9 I=1,NY TYPE0034
9 XSAM(I,LX)=X(1,J) TYPE0035
C XSAM=SAMPLE OF UP TO 100 CASES USED FOR INITIAL ESTIMATES OF PARAMETERS. TYPE0036
C 1=TERS. TYPE0037
WRITE TAPE 4,((X(1,J),1=1,MX),1=1,LONG) TYPE0038
IF (KRMN)10,1,7 TYPE0039
10 CALL INITIAL TYPE0040
IRM=IRM TYPE0041
C IRM=NUMBER OF TYPES ASSUMED. IF IRM=0, 6 DIFFERENT ANALYSES ARE DONE ASSUMING 1 TO 6 TYPES. TYPE0042
C IF (IRM)11,11,12 TYPE0043
11 IR=IR+1 TYPE0044
12 IR1=IR+1 TYPE0045
METH=MOTH TYPE0046

```

	XIR=IR	TYPE 0051
	KLX=(IR1+(IR1+1))/2	TYPE 0052
	JATMX = (MY1+(MX1+1)*IR)/2	TYPE 0053
C	PERS(K)=PROPORTION OF POPULATION OF TYPE K.	TYPE 0054
	DO 913 K=2,IR1	TYPE 0055
	IF (IR1) 13,13,912	TYPE 0056
912	IF (PERS(K)) 13,13,913	TYPE 0057
13	PERS(K)=1./XIR	TYPE 0058
913	CONTINUE	TYPE 0059
	ITER=0	TYPE 0060
	IDIV=0	TYPE 0061
	PROBA=-1040576.0005	TYPE 0062
14	ITER=ITER+1	TYPE 0063
	IF (DETERM(1)) 3500,3501,3501	TYPE 0064
3500	DETERM(1)=0.0	TYPE 0065
	GO TO 346	TYPE 0066
3501	CALL MOMENT	TYPE 0067
	WRITE OUTPUT TAPE 2,9916,ITER,IR,PROB	TYPE 0068
9916	FORMAT(10M ITERATION 13,15M, LIKELIHOOD OF 13,23M TYPES IN THIS SAT	TYPE 0069
	IMPLE RE10.0)	TYPE 0070
	IF (PHOB-PROBA) 346,347,347	TYPE 0071
346	IDIV=IDIV+1	TYPE 0072
	ITER=ITER-1	TYPE 0073
	WRITE OUTPUT TAPE 2,9915,ITERA	TYPE 0074
9915	FORMAT(10M ITERATION 13,9M DIVERGES)	TYPE 0075
	DO 3445 K=1,JATMX	TYPE 0076
3445	B(K)= 1.5*B(K)	TYPE 0077
	IF (IDIV-1) 115,3446,115	TYPE 0078
3446	DO 3447 K=1,JATMX	TYPE 0079
3447	B(K)= -B(K)	TYPE 0080
	GO TO 115	TYPE 0081
347	IDIV=0	TYPE 0082
	IF (METH) 344,348,348	TYPE 0083
344	IF (ITER-IDIMP) 348,348,3444	TYPE 0084
3444	METH=0-METH	TYPE 0085
C	IDUMP= THE NUMBER OF PRELIMINARY ITERATIONS BY SUCCESSIVE	TYPE 0086
C	SUBSTITUTIONS BEFORE NEWTON-RAPHSON ITERATION FOR METH=-1,-2,OR -3	TYPE 0087
148	PROBA=PROR	TYPE 0088
	IF (SENSE SWITCH 3) 349,249	TYPE 0089
149	WRITE OUTPUT TAPE 2,9910	TYPE 0090
9910	FORMAT(7M MOMENT)	TYPE 0091
	WRITE OUTPUT TAPE 2,9914,(((V(I,J),J=1,KLX),I=1,IJMX)	TYPE 0092
	WRITE OUTPUT TAPE 2,9914,(((COV(I,J,K),I=1,MX),J=1,MX),K=1,IR1)	TYPE 0093
	WRITE OUTPUT TAPE 2,9914,(((COV(I,J,K),I=1,MX),J=1,MX),K=1,IR1)	TYPE 0094
249	CONTINUE	TYPE 0095
114	CALL NEWTON	TYPE 0096
	IF (SENSE SWITCH 3) 350,250	TYPE 0097
150	WRITE OUTPUT TAPE 2,9911	TYPE 0098
9911	FORMAT(7M NEWTON)	TYPE 0099
	WRITE OUTPUT TAPE 2,9914,(((A(I,J),I=1,JATMX),J=1,JATMX)	TYPE 0100

```

      WRITE OUTPUT TAPE 2,9914,(H(I),I=1,JATMX)
9914 FORMAT(6F12.6)
250 CONTINUE
      IF(METH) 115,115,248
C      IF METH = n,SUCCESSIVE SUBSTITUTION IS USED,OTHERWISE
C      NEWTON-RAPHSON ITERATION.
248 CALL MATINV(A,JATMX,B,1,NA)
      IF(SENSE SWITCH3) 351,251
351 WRITE OUTPUT TAPE 2,9912
9912 FORMAT(7M MATINV)
      WRITE OUTPUT TAPE 2,9914,((A(I,J),I=1,JATMX),J=1,JATMX)
      WRITE OUTPUT TAPE 2,9914,(B(I),I=1,JATMX)
251 CONTINUE
115 CALL RAPHSN
      IF(IRUN) 15,16,15
C      IF IRUN=0,ONLY THE LAST ITERATION IS PRINTED.
19 CALL RESULT
16 IF(ITER-ITERM)17,19,19
17 DO 18 K=1,JATMX
      TA=B(K)
      TA=ABS(TA)
      IF(TA-CONV)10,18,14
18 CONTINUE
19 IF(IRUN) 21,20,21
20 CALL RESULT
21 CALL PLACE
      IF(IRM)22,22,1
22 IF(IRM=0)11,1,1
23 END FILE 2
      CALL EXIT
      END(0,1,0,0,0)

```

```

TYPE0101
TYPE0102
TYPE0103
TYPE0104
TYPE0105
TYPE0106
TYPE0107
TYPE0108
TYPE0109
TYPE0110
TYPE0111
TYPE0112
TYPE0113
TYPE0114
TYPE0115
TYPE0116
TYPE0117
TYPE0118
TYPE0119
TYPE0120
TYPE0121
TYPE0122
TYPE0123
TYPE0124
TYPE0125
TYPE0126
TYPE0127
TYPE0128
TYPE0129
TYPE0130
TYPE0131

```

```

SUBROUTINE INITIAL
DIMENSION V(5,3000),Z(6),PER(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(5TYPE0132
1,5,7),ALPHA(7),DETERM(7),G(7),IX(4),ID(4),V(126,26),A(126,126),B(1TYPE0133
226),BV(126),RECORD(12),SD(5),XSAM(5,100),MAT(100,100)
COMMON X,Z,PER,PERS,AV,COV,COVIN,ALPHA,DETERM,G,V,A,B,BV,RECORD,SDTYPE0134
1,MX,MX1,IR,XIR,IR1,IJMN,XLX,NY,XNX,JATMX,CONV,ITERM,LX,XSAM,MAT,ETTYPE0135
2PH,IX,ID,ITER,PROR,METH,IJUMP
EQUIVALENCE (XSAM,V),(A,MAT,X)
THIS SUBROUTINE DETERMINES INITIAL ESTIMATES OF THE MEANS AND COVATYPE0136
RIANCES OF THE TYPES BY APPLYING A CLUSTERING PROCEDURE TO A SAMPLTYPE0137
2E OF UP TO 1000 CASES.
FIRST THE SAMPLE MEANS AND COVARIANCES ARE DETERMINED.
X(L)=LX
DO 22 I=1,MX
AV(I,1)=0.
DO 1 K=1,LV
1 AV(I,1)=AV(I,1)+XSAM(I,K)
22 AV(I,1)=AV(I,1)/X(LX)
DO 3 J=1,MX
DO 3 J=1,MX
COV(I,J,1)=0.
DO 2 K=1,LV
2 COV(I,J,1)=COV(I,J,1)+XSAM(I,K)*XSAM(J,K)
COV(I,J,1)=COV(I,J,1)/XLV-AV(I,1)*AV(J,1)
3 COV(J,I,1)=COV(I,J,1)
DO 4 I=1,MX
AN LX BY LX MATRIX IS COMPUTED, AN ELEMENT CORRESPONDING TO A PAIRTYPE0138
1 OF POINTS IS 1 IF AND ONLY IF BOTH POINTS LIE WITHIN A BOX WHOSETYPE0139
2 SIDES ARE ONE STANDARD DEVIATION LONG.
SDV=COV(1,1,1)
4 SD(1)=SQRT(SDV)
DO 7 K=1,LV
DO 7 L=K,LV
DO 5 I=1,MX
R=XSAM(I,K)-XSAM(I,L)
R=ABS(R)
IF(R=SD(1))5,6,6
5 CONTINUE
MAT(L,K)=1
MAT(K,L)=1
GO TO 7
6 MAT(L,K)=0
MAT(K,L)=0
7 CONTINUE
DO 14 M=2,7
THE CENTROID OF A CLUSTER IS THE POINT WITH THE GREATEST NUMBER OFTYPE0140
1 OTHER POINTS IN A BOX AROUND IT.
FIRST THE INDEX OF THE POINT OF GREATEST DENSITY IS FOUND, THE TYPTYPE0141
16 MEANS ARE THE CO-ORDINATES OF THE DENSEST POINT.
MAX=0
DO 11 K=1,LX
LB=0
DO 11 L=1,LV

```

A LB=(M+MAT(1,M)	TYPE 0105
IF((M-MAX)1,1,9	TYPE 0106
0 MAX=LB	TYPE 0107
LAK=K	TYPE 0108
11 CONTINUE	TYPE 0109
DO 11 I=1,MX	TYPE 0110
AV(I,M)=XSAM(I,LAK)	TYPE 0111
DO 11 J=1,MX	TYPE 0112
COV(J,I,M)=COV(J,I,1)	TYPE 0113
11 COV(I,J,M)=COV(J,I,M)	TYPE 0114
DO 14 K=1, LX	TYPE 0115
IF(MAT(K,LAK)14,14,12	TYPE 0116
12 DO 13 L=1,LX	TYPE 0117
13 MAT(K,L)=L	TYPE 0118
C ALL PREVIOUSLY CLUSTERED POINTS ARE ERASED. THE CENTROID OF THE	TYPE 0119
C 1ST CLUSTER THUS WILL BE THE POINT WITH THE GREATEST NUMBER OF PREVIOUSLY	TYPE 0200
C 2IOUSLY UNCLUSTERED POINTS WITHIN A BOX AROUND IT.	TYPE 0201
14 CONTINUE	TYPE 0202
DO 15 I=1,MX	TYPE 0203
DO 15 J=1,MX	TYPE 0204
15 A(I,J)=COV(I,J,1)	TYPE 0205
CALL MATINV(A,MX,R,0,DA)	TYPE 0206
AD=ANSF(DA)/DA	TYPE 0207
DA=(SQRTF(AD/DA))*AD	TYPE 0208
DO 16 K=1,7	TYPE 0209
PERS(K)=1.0	TYPE 0210
DETERM(K)=DA	TYPE 0211
DO 16 I=1,MX	TYPE 0212
DO 16 J=1,MX	TYPE 0213
16 COV(I,J,K)=A(I,J)	TYPE 0214
C THE ROUTINE ALSO READS IN INITIAL ESTIMATES UNTIL A BLANK CARD	TYPE 0215
C IS ENCOUNTERED.	TYPE 0216
17 READ INPUT TAPE 3,122,K,DA	TYPE 0217
122 FORMAT(14,F4.4)	TYPE 0218
IF(K) 18,21,18	TYPE 0219
18 K=K+1	TYPE 0220
PERS(K)=DA	TYPE 0221
READ INPUT TAPE 3,123,(AV(I,K),I=1,MX)	TYPE 0222
READ INPUT TAPE 3,123,(SN(I),I=1,MX)	TYPE 0223
23 FORMAT(9F4.4)	TYPE 0224
DO 19 J=1,MX	TYPE 0225
READ INPUT TAPE 3,123,(COV(I,J,K),I=1,MX)	TYPE 0226
DO 19 I=1,MX	TYPE 0227
COV(I,J,K)=COV(I,J,K)+SD(I)*SD(J)	TYPE 0228
19 A(I,J)=COV(I,J,K)	TYPE 0229
CALL MATINV(A,MX,R,0,DA)	TYPE 0230
AD=ANSF(DA)/DA	TYPE 0231
DA=(SQRTF(AD/DA))*AD	TYPE 0232
DETERM(K)=DA	TYPE 0233
DO 20 I=1,MX	TYPE 0234
DO 20 J=1,MX	TYPE 0235
20 COV(I,J,K)=A(I,J)	TYPE 0236
GO TO 17	TYPE 0237
21 RETURN	TYPE 0238
END(1,0,n,0)	TYPE 0239

```

SUBROUTINE DENSITY                                     TYPE0240
  DIMENSION Y(5,3000),Z(5),PER(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(5TYPE0241
1,5,7),ALPHA(7),DETERM(7),G(7),IX(4),ID(4),V(126,28),A(126,126),B(1TYPE0242
226),RV(126),RECORD(12),SN(5),XSAM(5,100),MAT(100,100)      TYPE0243
  COMMON X,Z,PER,PERS,AV,COV,COVIN,ALPHA,DETERM,G,V,A,B,RV,RECORD,SDTYPE0244
1,MX,MX1,IR,XIR,IR1,IJMNX,KLX,NX,XNX,JATMX,CONV,ITER,LY,XSAM,MAT,ETYPE0245
2PH,IX,ID,ITER,PROR,METH,TDUMP                               TYPE0246
  EQUIVALENCE (XSAM,V),(A,MAT,Y)                             TYPE0247
C   FOR CURRENT ESTIMATES OF THE PARAMETERS, THIS ROUTINE COMPUTES- TYPE0248
C   ALPHA(K)=MULTIVARIATE NORMAL DENSITY FOR TYPE K AT POINT Z. TYPE0249
C   EPH=DENSITY AT Z FOR THE MIXTURE OF DISTRIBUTIONS          TYPE0250
C   G(K)=ALPHA/EPH AT POINT Z.                                  TYPE0251
  DO 3 K=2,IR1                                               TYPE0252
    AL=0.                                                       TYPE0253
    DO 2 I=1,MV                                                TYPE0254
      IA=I+1                                                    TYPE0255
      DO 1 J=1,MV                                                TYPE0256
1    AL=AL-COVIN(J,I,K)*(Z(IA)-AV(I,K))*(Z(J+1)-AV(J,K))      TYPE0257
2    AL=AL+0.5*COVIN(I,I,K)*(Z(IA)-AV(I,K))**2                TYPE0258
3    ALPHA(K)=DETERM(K)*EXP(AL)                                  TYPE0259
    EPH=0.                                                       TYPE0260
    DO 4 K=2,IR1                                                TYPE0261
4    EPH=EPH+PERS(K)*ALPHA(K)                                    TYPE0262
    DO 5 K=2,IR1                                                TYPE0263
5    G(K)=ALPHA(K)/EPH                                           TYPE0264
    G(1)=1.                                                       TYPE0265
C   THE ADDITION OF A DUMMY TYPE OF DENSITY G=1 SIMPLIFIES THE COMPUTATYPE0266
C   ION OF LOWER ORDER MOMENTS WITH ONE OR MORE TYPES OMITTED. TYPE0267
  RETURN                                                         TYPE0268
  END(0.1,0.0,0.0)                                             TYPE0269

```

```

SUBROUTINE MOMENT
DIMENSION Y(5,300),Z(6),PER(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(5TYPE0270
1,5,7),ALPHA(7),DETERM(7),G(7),IX(4),ID(4),V(126,28),A(126,126),B(1TYPE0272
226),HV(126),RECORD(12),SN(5),XSAM(5,100),MAT(100,100) TYPE0273
COMMON X,7,PER,PERS,AV,COV,COVIN,ALPHA,DETERM,G,V,A,B,HV,RECORD,SDTYPE0274
1,MX,MX1,IR,XIR,IR1,IJMN,KLX,NY,XNX,JAT=X,CONV,ITERM,LX,XSAM,MAT,ETTYPE0275
2PM,IX,ID,ITER,PROB,METH,IDUMP TYPE0276
EQUIVALENCE (XSAM,V),(A,MAT,X) TYPE0277
C THIS ROUTINE COMPUTES THE GENERALIZED MOMENTS V(IJMN,KL) FOR VARIATTYPE0278
C 10LES I,J,M, AND N, AND TYPES K AND L. THE MOMENTS REALLY HAVE SIX TYPE0279
C 2 INDICES BUT ARE STORED AS A 2-DIMENSIONAL MATRIX. TRIANGULAR INDEXTYPE0280
C 3ING IS USED TO ELIMINATE DUPLICATION. TYPE0281
DO 1 IJMN=1,IJMNX TYPE0282
DO 1 KL=1,KLX TYPE0283
1 V(IJMN,KL)=0. TYPE0284
KRMN=0 TYPE0285
C INITIALIZE TYPE0286
REWIND 4 TYPE0287
PROB=XNX TYPE0288
DO 10 K=2,IR1 TYPE0289
10 PROB=PROB-XNX=PERS(K) TYPE0290
2 CALL BLOCK(NY,300,KRMN,KST,KEND,LONG) TYPE0291
READ TAPE 4,((X(I,J),I=1,MX),J=1,LONG) TYPE0292
C TAPE 4 IS READ IN BLOCKS OF 300 CASES. TYPE0293
DO 17 KA=1,LONG TYPE0294
Z(1)=1.0 TYPE0295
C THE ADDITION OF A DUMMY VARIABLE=1 SIMPLIFIES THE COMPUTATION OF LTYPE0296
C 1 OVER ORDER MOMENTS OMITTING ONE OR MORE VARIABLES. TYPE0297
DO 3 I=1,MX TYPE0298
3 Z(I+1)=X(I,KA) TYPE0299
CALL DENSITY TYPE0300
C PROB=LINSLIHOOD OF SAMPLE TYPE0301
IF(6PM)7,7,8 TYPE0302
7 INDEX=KST+KA-1 TYPE0303
WRITE OUTPUT TAPE 2,120,INDEX,6PM TYPE0304
180 FORMAT(49H NEGATIVE PROBABILITY DENSITY FOR OBSERVATION 19,3H = TYPE0305
1F16.9) TYPE0306
GO TO 9 TYPE0307
8 PROB=PROB+LOGF(6PM) TYPE0308
9 KL=0 TYPE0309
IF(METH) 19,12,11 TYPE0310
11 DO 4 K=1,IR1 TYPE0311
DO 4 L=1,K TYPE0312
KL=KL+1 TYPE0313
GT1=G(K)=G(L) TYPE0314
IJMN=0 TYPE0315
DO 4 I=1,MX1 TYPE0316
GT2=GT1+Z(I) TYPE0317
DO 4 J=1,I TYPE0318
GT3=GT2+Z(J) TYPE0319

```

```

      DO 4 M=1,N
      GT=GT+7*(M)
      DO 4 N=1,M
      IJMN=IJMN+1
      V(IJMN,KL)=V(IJMN,KL)+GT*7*(M)
17 CONTINUE
15 IF(KHMM)D,5,2
      DO 4 IJMN=1,IJMN*
      DO 4 KL=1,KL*
      V(IJMN,KL)=V(IJMN,KL)/XNY
      REIJMN
12 KL=
      DO 13 K=1,IH1
      FOR MATHS,WE DO NOT NEED ALL THE MOMENTS
      IJMN=2
      DO 14 J=2,NY1
      GT1=Z(IJ)=G(K)
      JS=
      DO 14 J=1,I
      V(IJMN,KL)=V(IJMN,KL)+GT*7*(J)
      JS=JS+J
14 IJMN=IJMN+JS
      DO 13 L=1,K
      V(I,AL)=V(I,KL)+G(K)*G(L)
13 KL=KL+1
      GO TO 17
      END( .1, .1, .1 )

```

```

TYPE0320
TYPE0321
TYPE0322
TYPE0323
TYPE0324
TYPE0325
TYPE0326
TYPE0327
TYPE0328
TYPE0329
TYPE0330
TYPE0331
TYPE0332
TYPE0333
TYPE0334
TYPE0335
TYPE0336
TYPE0337
TYPE0338
TYPE0339
TYPE0340
TYPE0341
TYPE0342
TYPE0343
TYPE0344
TYPE0345
TYPE0346

```

```

FUNCTION U(JA,JA,MA,NA,KA,LA)                                TYPE0347
DIMENSION X(5,100),Z(6),PER(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(5TYPE0348
1,5,7),ALPHA(7),DETERM(7),G(7),IX(4),ID(4),V(126,28),A(126,126),B(1TYPE0349
/26),MV(126),RECORD(12),SH(5),XSAM(5,100),MAT(1,1,10,1)      TYPE0350
COMMON X,Z,PER,PERS,AV,COV,COVIN,ALPHA,DETERM,G,V,A,H,MV,RECORD,SDTYPE0351
1,MX,MX1,IR,XIR,IR1,IJMN,KLX,NY,XNX,JATMX,CONV,ITERM,LV,XSAM,MAT,ETTYPE0352
2PH,IX,ID,ITER,PROB,METH,IDUMP                                TYPE0353
EQUIVALENCE (XSAM,V),(A,MAT,X)                                TYPE0354
C THIS ROUTINE FINDS THE MOMENT CORRESPONDING TO 6 INDICES BY COMPUTTYPE0355
C ING THE 2 INDICES OF THE MOMENT IN THE MATRIX V.            TYPE0356
IX(1)=JA                                                        TYPE0357
IX(2)=JA                                                        TYPE0358
IX(3)=MA                                                        TYPE0359
IX(4)=NA                                                        TYPE0360
C THE FIRST 4 INDICES ARE PUT IN ORDER FROM SMALLEST TO LARGEST TYPE0361
DO 3 J=1,4                                                       TYPE0362
KAT=100                                                         TYPE0363
DO 2 I=1,4                                                       TYPE0364
IF(KAT-IX(I)) 2,2,1                                             TYPE0365
1 KAT=IX(I)                                                     TYPE0366
K=I                                                             TYPE0367
GO TO 2                                                         TYPE0368
2 CONTINUE                                                       TYPE0369
IX(K)=IX(I)                                                     TYPE0370
3 ID(J)=KAT                                                      TYPE0371
C THE LAST 2 INDICES ARE PUT IN ORDER FROM LARGER TO SMALLER TYPE0372
IF (KA-LA)4,5,5                                                 TYPE0373
4 KAT=LA                                                         TYPE0374
LAT=KA                                                         TYPE0375
GO TO 6                                                         TYPE0376
5 KAT=KA                                                         TYPE0377
LAT=LA                                                         TYPE0378
6 IJMN=1                                                         TYPE0379
C THE APPROPRIATE INDICES OF THE MOMENT MATRIX V ARE COMPUTED. TYPE0380
DO 8 I=1,4                                                       TYPE0381
LPR=1                                                           TYPE0382
DO 7 J=1,1                                                       TYPE0383
7 LPR=(LPR+(ID(I)-2+J))/J                                       TYPE0384
8 IJMN=IJMN+LPR                                                 TYPE0385
KL=(KAT+(KAT-1))/2+LAT                                         TYPE0386
U=V(IJMN,KI)                                                   TYPE0387
RETURN                                                         TYPE0388
END(0,1,0,0,0)                                                 TYPE0389

```

```

      FUNCTION ONE(I,J)
      THIS IS JUST THE KRONCKER DELTA
      IF(I=J) ONE=1,2,3
1     ONE= .
      GO TO 3
2     ONE=1.
      RETURN
      END

```

```

TYPE=390
TYPE=391
TYPE=392
TYPE=393
TYPE=394
TYPE=395
TYPE=396
TYPE=397

```

```

FUNCTION FM(I1,I2,IA,IS,IP)                                TYPE=398
DIMENSION V(5,1),Z(6),PER(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(5TYPE=399
1,5,7),ALPHA(7),DETERM(7),G(7),IX(4),ID(4),U(126,28),A(126,126),B(1TYPE=400
226),HV(126),RECORD(12),SD(5),XSAM(5,1),MAT(1,1)          TYPE=401
COMMON X,7,PER,PERS,AV,COV,COVIN,ALPHA,DETERM,G,V,A,H,HV,RECORD,SDTYPE=402
1,MX,MX1,IP,XIH,IN,IUMX,KLX,NV,XKX,JATMX,CONV,ITER,LX,XSAM,MAT,ETTYPE=403
2PM,IX,ID,ITER,PROR,METH,TDMP                                TYPE=404
EQUIVALENCE (XSAM,V),(A,MAT,X)                               TYPE=405
C   FM IS A TERM WHICH APPEARS IN THE PARTIAL DERIVATIVE OF THE MAXIMUITYPE=406
C   LIKELIHOOD EQUATIONS WITH RESPECT TO THE TYPE MEANS.    TYPE=407
ENB=0                                                         TYPE=408
IA1=IA-1                                                       TYPE=409
DO 1 I=1,MV                                                    TYPE=410
IB=1                                                            TYPE=411
1 ENB=FM+COVIN(IA,IP,IP)=0(1,1,1,1,IB+1,IP,IS)-AV(IH,IP)+U(1,1,1,1,ITYPE=412
1J,IP,IS)                                                       TYPE=413
RETURN                                                         TYPE=414
END(L,1,1,1,1)                                                 TYPE=415

```



```

FUNCTION DERV(1,10,15,1A,1R,1P)
DIMENSION V(5,10),Z(1,2(6),PERS(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(51TYPE=434
1,5,7),ALPHA(7),DETERM(7),G(7),IX(4),ID(4),V(120,20),A(120,100),R(1TYPE=435
22A),-V(12A),RECORD(12),SP(5),XSAM(5,10),MAT(1 1 1) TYPE=437
COMMON X,7,PEN,PERS,AV,COV,CVIN,ALPHA,DETERM,G,V,A,H,-V,RECORD,SDTYPE=438
1,MV,X1,1R,X1R,1R1,1JMX,KX,NV,YNX,1ATMX,COV,1TEN,LY,XRAM,MAT,ETTYPE=439
2PH,1Y,1D,1TR,PROR,METH,INOM
EQUIVALENCE (XSAM,V),(A,MAT,V)
THIS ROUTINE COMPUTES THE DERIVATIVE OF THE IJS MOMENT WITH
C RESPECT TO THE AMP PARAMETER.
C
IF(1A=1)1,1,2
1 DERV = -U(1,10,1,1,15,1P)
GO TO 1
2 IF(1R=1)3,1,6
3 DERV=PEPS(1P)=EM(1,10,1A,1S,1P)
IF(1P=1S)5,4,6
4 DERV=DERV+EM(1,10,1A,1,1P)
GO TO 1
5 DERV=PEPS(1P)=SIG(1,1,1A,1R,1S,1P)
IF(1P=1S)8,7,8
7 DERV=DERV+SIG(1,10,1A,1R,1,1P)
8 IF(1A=10)10,9,10
9 DERV=0.5*DERV
10 RETURN
END

```

```

SUBROUTINE NEWTON
DIMENSION V(5,40),PFR(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(5,5,7),ALPHA(7),DETERM(7),G(7),IX(4),ID(4),V(126,28),A(126,126),U(126,126),H(126,126),RECORR(12),SR(5),XRAM(5,10),MAT(10,10)
COMMON X,Z,PFR,PERS,AV,COV,COVIN,ALPHA,DETERM,G,V,A,H,HV,RECORR,SDTYPE
1,MV,MX1,IR,XIR,IR1,IJMN,KLX,NX,XX,JATMX,CONV,ITER,LX,XRAM,MAT,ET
2PM,IX,ID,ITER,PROR,METH,IDUMF
EQUIVALENCE (XRAM,V),(A,MAT,X)
THIS ROUTINE SETS UP THE NEWTON-RAPHSON MATRIX EQUATION FOR THE
RANGES IN THE PARAMETERS FOR THE NEXT ITERATION.
METH=XARSF(METH)
METH = +1 OR -1 IMPLIES SIMPLIFIED MAXIMUM-LIKELIHOOD EQUATIONS
METH = +2 OR -2 IMPLIES COMPLETE MAXIMUM-LIKELIHOOD EQUATIONS
METH = +3 OR -3 IMPLIES COMPLETE EQUATIONS DIVIDED BY UDS
IF(METH)42,42,4
IAT=1
DO 27 K=2,IR1
UJS=U(1,1,1,1,K,1)
DO 27 J=1,MX1
UJS=U(1,1,1,J,K,1)
DO 27 I=J,MX1
UIS=U(1,1,1,I,K,1)
JAT=1
IAT=IAT+1
DO 27 L=2,IR1
DO 27 N=1,MX1
DO 27 M=N,MX1
JAT=JAT+1
TEMP=DERV(I,J,K,M,N,L)
IF(I=1) 27,27,0
GO TO (12,13,14),M,K
11 TEMP=TEMP-AV(I-1,K)=DERV(J,1,K,M,N,L)
IF(J=1)12,12,11
12 TEMP=TEMP-(COV(I-1,J-1,K)=AV(I-1,K)=AV(J-1,K)=DERV(1,1,K,M,N,L)
1- AV(J-1,K)=DERV(1,1,K,M,N,L)
13 IF(K=L)27,13,27
14 IF(N=1)14,14,17
15 GO TO (16,15,15),M,K
16 TEMP=TEMP-ONE(I,M)=UJS-(1-ONE(J,1))=ONE(J,M)=UIS-UOS=
1(ONE(I,M)=AV(J-1,K)=ONE(I,M)=AV(I-1,K))
GO TO 27
17 TEMP=TEMP-ONE(J,1)=ONE(I,M)-(1-ONE(J,1))=ONE(I,M)=AV(J-1,K)=
1ONE(J,M)=AV(I-1,K)
GO TO 27
18 IF(J=1)27,27,18
19 TEMP=COV(I-1,M-1,L)=COV(J-1,M-1,L)+COV(I-1,M-1,L)=COV(J-1,M-1,L)
GO TO (2,10,10),M,K
20 TEMP=TEMP-UOS
21 IF(M=N)22,21,22
22 TEMP=.5*TEMP
23 TEMP=TEMP+TEMP
24 A(IAT,JAT)=TEMP
L=
DO 33 K=2,IR1
L=L+1

```

```

      UOS=U(1,1,1,1,K,1)
      B(L)=1.-UOS
      DO 29 I=2,MX1
      L=L+1
      B(L)=AV(I-1,K)
      GO TO (29,28,28),MEK
28  B(L)=B(L)+UOS
29  B(L)=B(L)-U(1,1,1,1,K,1)
      DO 33 J=2,MX1
      DO 33 I=J,MX1
      L=L+1
      GO TO (31,3,3),MEK
30  B(L)=UOS*(COV(I-1,J-1,K)-AV(I-1,K)+AV(J-1,K))+AV(I-1,K)+
      U(1,1,1,J,K,1)+AV(J-1,K)+U(1,1,1,I,K,1)
      GO TO 33
31  B(L)=COV(I-1,J-1,K)+AV(I-1,K)+AV(J-1,K)
33  B(L)=B(L)-U(1,1,1,J,K,1)
      GO TO (37,37,34),MEK
34  IAT=0
      DO 36 K=2,IR1
      JAT=0
      DO 35 L=2,IR1
      DO 35 N=1,MX1
      DO 35 M=N,MX1
      JAT=JAT+1
35  BV(JAT)=DERV(1,1,K,M,N,L)/U(1,1,1,1,K,1)
      DO 36 J=1,MX1
      DO 36 I=J,MX1
      IAT=IAT+1
      JAT=0
      DO 36 L=2,IR1
      DO 36 N=1,MX1
      DO 36 M=N,MX1
      JAT=JAT+1
36  A(IAT,JAT)=A(IAT,JAT)+B(IAT)*BV(JAT)
37  RETURN
42  DO 44 I=1,IR
      DO 43 J=1,I
      A(I,J)=U(1,1,1,1,I+1,J+1)
43  A(J,I)=A(I,J)
44  B(I)=U(1,1,1,1,I,1)-1.0
      CALL MATINV(A,IR,B,1,BA)
      L=IR
      DO 52 K=2,IR1
      PERT=U(1,1,1,1,K)
      DO 52 I=1,MX
      L=L+1
      B(L)=U(1,1,1,I+1,K)/PERT-AV(I,K)
      BV(I)=AV(I,K)+B(L)
      DO 52 J=1,I
      L=L+1
52  B(L)=U(1,1,J+1,I+1,K)/PERT-BV(I)+RV(J)-COV(I,J,K)
      GO TO 37
      END

```

```

TYPE=514
TYPE=515
TYPE=516
TYPE=517
TYPE=518
TYPE=519
TYPE=520
TYPE=521
TYPE=522
TYPE=523
TYPE=524
TYPE=525
TYPE=526
TYPE=527
TYPE=528
TYPE=529
TYPE=530
TYPE=531
TYPE=532
TYPE=533
TYPE=534
TYPE=535
TYPE=536
TYPE=537
TYPE=538
TYPE=539
TYPE=540
TYPE=541
TYPE=542
TYPE=543
TYPE=544
TYPE=545
TYPE=546
TYPE=547
TYPE=548
TYPE=549
TYPE=550
TYPE=551
TYPE=552
TYPE=553
TYPE=554
TYPE=555
TYPE=556
TYPE=557
TYPE=558
TYPE=559
TYPE=560
TYPE=561
TYPE=562
TYPE=563
TYPE=564
TYPE=565
TYPE=566
TYPE=567

```

```

SUBROUTINE RAPHSON
  DIMENSION V(5,3000),Z(6),PER(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(5TYPE0560
1,5,7),ALPHA(7),DETERM(7),G(7),IX(4),ID(4),V(126,28),A(126,126),B(1TYPE0570
226),BV(126),RECORD(12),SR(5),XCAM(5,100),MAT(100,100)
COMMON X,7,PER,PERS,AV,COV,COVIN,ALPHA,DETERM,G,V,A,B,MV,RECORD,SDTYPE0572
1,MX,MX1,IR,XIR,IR1,IJMN,KLX,NY,XY,XNY,JATMX,CONV,ITERM,LX,XCAM,MAT,ETTYPE0573
2PM,IX,ID,ITER,PROG,METN,TDUMP
EQUIVALENCE (XCAM,V),(A,MAT,X)
THIS ROUTINE ADDS THE APPROPRIATE CHANGES TO THE PARAMETERS TO OBTTYPE0574
1AIN THEIR VALUES FOR THE NEXT ITERATION.
AKAT=2.
IF(METN)30,30,23
NEWTON=RAPHSON ITERATION
23 IAT=1
MXT=(MX1-(MX1+1))/2
SHORTEN INCREMENT VECTOR UNTIL ALL PERCENTAGES ARE WITHIN BOUNDSTYPE0575
DO 10 K=2,IR1
AKAZ=PERS(K)/(2.-B(IAT))
IF(AKAZ)7,7,A
7 AKAZ=AKAZ+.5/M(IAT)
8 IF(AKAZ=AKAT)9,1,10
9 AKAT=AKAZ
10 IAT=IAT+MXT
IF(AKAT=1.0)11,13,13
11 DO 12 K=1,JATMX
12 B(K)=AKAT*B(K)
13 IAT=0
DO 20 K=2,IR1
IAT=IAT+1
PERS(K)=PERS(K)+B(IAT)
DO 24 I=1,MV
IAT=IAT+1
24 AV(I,K)=AV(I,K)+B(IAT)
DO 25 J=1,MX
DO 25 I=J,MX
IAT=IAT+1
COVIN(I,J,K)=COVIN(I,J,K)+B(IAT)
COVIN(J,I,K)=COVIN(I,J,K)
A(I,J)=COVIN(I,J,K)
25 A(J,I)=A(I,J)
CALL MATINV(A,MX,MV,100)
AD=AMSP(100)/100
DETERM(K)=DETERM(AD=100)=AD
IF(A)27,27,22
27 INDEFIN=1
WRITE OUTPUT TAPE 2,121,INDEF,NA
221 FORMAT(44H DETERMINANT OF COVARIANCE INVERSE FOR TYPE 12,1MF1A,9)TYPE0614
DETERM(1)=1.
DO 26 J=1,MX
DO 26 I=J,MX
26 COV(I,J,K)=A(I,J)
26 RETURN
TYPE0615
TYPE0616
TYPE0617
TYPE0618
TYPE0619

```

C	SUCCESSIVE SUBSTITUTIONS.	TYPE 0620
31	L=IR	TYPE 0621
	MYT=1	TYPE 0622
C	SHORTEN INCREMENT VECTOR UNTIL ALL PERCENTAGES ARE WITHIN BOUNDS	TYPE 0623
	DO 36 K=2,IR1	TYPE 0624
	AKA7=-PERS(K)/(2.*R(K-1))	TYPE 0625
	IF(AKA7)37,33,34	TYPE 0626
33	AKA7 = AKA7 + .5/R(K-1)	TYPE 0627
34	IF(AKA7-AKAT)35,34,36	TYPE 0628
35	AKAT=AKA7	TYPE 0629
36	CONTINUE	TYPE 0630
	IF(AKAT-1.)37,39,39	TYPE 0631
37	DO 38 K=1,IR	TYPE 0632
38	U(K)=AKAT*R(K)	TYPE 0633
39	DO 55 K=2,IR1	TYPE 0634
	PERS(K)=PERS(K)+ R(K-1)	TYPE 0635
	DO 52 J=1,MX	TYPE 0636
	L=L+1	TYPE 0637
47	AV(I,K)=AV(I,K)+R(L)	TYPE 0638
	DO 52 J=1,I	TYPE 0639
	L=L+1	TYPE 0640
49	COV(I,J,K)=COV(I,J,K)+R(L)	TYPE 0641
51	COV(J,I,K)=COV(I,J,K)	TYPE 0642
	A(I,J)=COV(I,J,K)	TYPE 0643
	A(J,I)=A(I,J)	TYPE 0644
52	CONTINUE	TYPE 0645
53	CALL MATINV(A,MX,BV,n,DA)	TYPE 0646
	AD=ADSF(DA)/DA	TYPE 0647
	DETERM(K)=(SORTF(AD/DA))=AD	TYPE 0648
	DO 54 I=1,MX	TYPE 0649
	DO 54 J=1,MX	TYPE 0650
54	COV(I,J,K)=A(I,J)	TYPE 0651
55	CONTINUE	TYPE 0652
	GO TO 29	TYPE 0653
	END	TYPE 0654

```

SUBROUTINE RESULT
  DIMENSION V(5,300),7(4),PER(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(5TYPE0655
1,5,7),ALPHA(7),DETERM(7),G(7),IX(4),ID(4),V(126,28),A(126,126),B(1TYPE0656
226),MV(126),RECORD(12),SD(5),XSAM(5,100),MAT(100,100) TYPE0657
COMMON X,7,PER,PERS,AV,COV,COVIN,ALPHA,DETERM,G,V,A,M,MV,RECORD,SDTYPE0658
1,MX,MX1,IR,XIR,IR1,IMNX,KLX,NY,MNX,JATMX,CONV,ITERM,LV,XSAM,MAT,ETYPE0659
2PH,IX,ID,ITER,PROR,METH,IDUMP TYPE0660
EQUVALENCE (XSAM,V),(A,MAT,X) TYPE0661
THIS ROUTINE PRINTS OUT THE PARAMETERS DESCRIBING EACH TYPE. TYPE0662
1 FORMAT(1M,15X,3AMAXIMUM-LIKELIKHOOD ANALYSIS OF TYPES//4Y,11A6) TYPE0663
2 FORMAT(1M,15X,35HCHARACTERISTICS OF THE WHOLE SAMPLE///3X) TYPE0664
3 FORMAT(3X/////14X,23HCHARACTERISTICS OF TYPE 14///3X) TYPE0665
4 FORMAT(11X,49HTHE PROPORTION OF THE POPULATION FROM THIS TYPE OF TYPE0666
13/3X) TYPE0667
5 FORMAT(3X,5HMEANS/5F12.2) TYPE0668
6 FORMAT(3X/23X,10HSTANDARD DEVIATIONS /5F12.2) TYPE0669
7 FORMAT(3X/23X,12HCORRELATIONS) TYPE0670
8 FORMAT (5F12.4) TYPE0671
9 FORMAT(3X/////10X,13HSAMPLE SIZE =11/10X,21HNUMBER OF VARIABLES =TYPE0672
112/10X,17HNUMBER OF TYPES =14/25X,14HITERATION NUMBER 13/3X, TYPE0673
21HLIKELIHOOD OF 12,23H TYPES IN THIS SAMPLE OF1A.8) TYPE0674
DO 10 J=1,MX TYPE0675
  JA=J+1 TYPE0676
  AV(J,1)=0(1,1,JA,1,1) TYPE0677
  DO 20 I=1,MX TYPE0678
    JA=J+1 TYPE0679
    DO 20 I=J,MX TYPE0680
      IA=I+1 TYPE0681
      COV(I,J,1)=0(1,1,JA,IA,1,1)-AV(J,1)*AV(I,1) TYPE0682
2    COV(J,I,1)=COV(I,J,1) TYPE0683
    WRITE OUTPUT TAPE 2,1,(RECORD(I),I=1,11) TYPE0684
    WRITE OUTPUT TAPE 2,9,MX,MX,IR,ITER,IR,PROR TYPE0685
    DO 14 K=1,12 TYPE0686
      K1=K+1 TYPE0687
      IF (K) 11,11,12 TYPE0688
11    WRITE OUTPUT TAPE 2,2 TYPE0689
      GO TO 13 TYPE0690
12    WRITE OUTPUT TAPE 2,3,K1 TYPE0691
      WRITE OUTPUT TAPE 2,4,PERS(K) TYPE0692
13    WRITE OUTPUT TAPE 2,5,(AV(I,K),I=1,MX) TYPE0693
      DO 14 I=1,MX TYPE0694
        SDV=COV(I,I,K) TYPE0695
        AD=AMRF(SDV)/SNV TYPE0696
        SDV=(SQRTF(AD*QDV))=AD TYPE0697
14    SN(I)=SDV TYPE0698
      DO 19 I=1,MX TYPE0699
        DO 19 J=1,MX TYPE0700
          A(I,J)=COV(I,J,K)/(SN(I)*SN(J)) TYPE0701
15    A(J,I)=A(I,J) TYPE0702
      WRITE OUTPUT TAPE 2,6,(SN(I),I=1,MX) TYPE0703
      WRITE OUTPUT TAPE 2,7 TYPE0704
      DO 16 J=1,MX TYPE0705
16    WRITE OUTPUT TAPE 2,8,(A(I,J),I=1,MX) TYPE0706
      RETURN TYPE0707
      ENDE 21, 100,1) TYPE0708

```

	ROUTINE PLACE	TYPE n710
	DIMENSION Y(5,1:10),Z(6),PER(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(5)TYPE n711	
	1,5,7),ALPHA(7),DETERM(7),G(7),IX(4),ID(4),A(126,28),A(126,126),B(1)TYPE n712	
	22A),HVE(7A),RECORD(12),SD(5),XSAM(5,100),MAT(1:10,1:1)TYPE n713	
	COMMON X,7,PER,PERS,AV,COV,COVIN,ALPHA,DETERM,G,V,A,H,HV,RECORD,SDTYPE n714	
	1,MX,4X1,IP,XIR,IR,1JMX,K(X,AV,XNX,JATMX,CONV,ITER,LV,XSAM,MAT,ETYPE n715	
	2PM,IX,ID,ITER,PROR,METH,IDLMP	TYPE n716
	EQUIVALENCE (XSAM,V),(A,MAT,X)	TYPE n717
C	THIS ROUTINE PRINTS OUT THE PROBABILITIES OF TYPE MEMBERSHIP FOR ETYPE n718	
C	EACH OBSERVATION	TYPE n719
	1 FORMAT(1X,12HPROGRAM: PROBS OF TYPE MEMBERSHIP//X,718)	TYPE n720
	2 FORMAT(1X,7F8.3)	TYPE n721
	WRITE OUTPUT TAPE 2,1,(1,1=1,10)	TYPE n722
	K0=0	TYPE n723
	KRMN=0	TYPE n724
	REWIND 4	TYPE n725
	3 CALL BLOCK(MX,1,0,KRMN,KST,REND,1000)	TYPE n726
	READ TAPE 4, ((X(I,J)),1=1,MX),1=1,1000)	TYPE n727
	DO 4 K=1,1000	TYPE n728
	DO 4 I=1,MX	TYPE n729
	4 Z(I+1)=X(I,K)	TYPE n730
	CALL DENSITY	TYPE n731
	KRMN=1	TYPE n732
	DO 5 I=2,101	TYPE n733
	5 G(I)=G(I)+PERS(I)	TYPE n734
	6 WRITE OUTPUT TAPE 2,2,K0,(G(I)),I=2,101)	TYPE n735
	IF(KRMN)7,7,3	TYPE n736
	7 RETURN	TYPE n737
	END(=1,0,0,0)	TYPE n738

	SUBROUTINE LOCKINX(LRLK,KRMN,KST,KEND,LONG)	TYPE=739
C	THIS ROUTINE COMPUTES NUMBERS USEFUL IN THE CONTROL OF THE INPUT	TYPE=740
C	AND OUTPUT OF LISTS OF LENGTH NX IN BLOCKS OF LENGTH LRLK.	TYPE=741
C	KRMN=NUMBER OF ITEMS REMAINING IN THE LIST	TYPE=742
C	LONG=LENGTH OF CURRENT BLOCK	TYPE=743
C	KST AND KEND ARE THE STARTING AND ENDING INDICES FOR THE ITEMS IN	TYPE=744
C	THE CURRENT BLOCK.	TYPE=745
	IF (KRMN)2,1,2	TYPE=746
1	KRMN=NX	TYPE=747
	KEND=0	TYPE=748
2	LONG=LRLK	TYPE=749
	KST=KEND+1	TYPE=750
	IF (KRMN-LRLK)3,4,4	TYPE=751
3	LONG=KRMN	TYPE=752
4	KEND=KEND+LONG	TYPE=753
	KRMN=KRMN-1 LONG	TYPE=754
	RETURN	TYPE=755
	END	TYPE=756

	SUBROUTINE MATINV(A,N,B,M,DETERM)	TYPE0757
	DIMENSION IPIVOT(126),A(126,126),B(126,1),INDEX(126,2),PIVOT(126,	TYPE0758
	EQUIVALENCE (IROW,JROW), (ICOLUMN,JCOLUMN), (AMAX, I, SWAP)	TYPE0759
C		TYPE0760
C	PROGRAMMED BY BUNTON S. GARBOW, ARGONNE NATIONAL LABORATORY,	TYPE0761
C	AND REPORTED IN IBM 7 4-719 SHARE LIBRARY AS AN F4 2.	TYPE0762
C	THIS SUBROUTINE COMPUTES THE INVERSE AND DETERMINANT OF	TYPE0763
C	MATRIX A, OF ORDER N, BY THE GAUSS-JORDAN METHOD. A-INVERSE	TYPE0764
C	REPLACES A, AND THE DETERMINANT OF A IS PLACED IN DETERM. IF	TYPE0765
C	M = 1 THE VECTOR B CONTAINS THE CONSTANT VECTOR WHEN MATINV IS	TYPE0766
C	CALLED, AND THIS IS REPLACED WITH THE SOLUTION VECTOR. IF M = 0,	TYPE0767
C	NO SIMULTANEOUS EQUATION SOLUTIONS ARE CALLED FOR, AND M IS NOT	TYPE0768
C	PERTINENT. N IS NOT TO EXCEED 50.	TYPE0769
C	A, N, B, M, AND DETERM IN THE ARGUMENT LIST ARE DUMMY VARIABLES.	TYPE0770
C		TYPE0771
C		TYPE0772
C	INITIALIZATION	TYPE 773
	1. DETERM=1.0	TYPE0774
	15 DO 20 J=1,N	TYPE0775
	20 IPIVOT(J)=1	TYPE0776
	30 DO 50 I=1,N	TYPE0777
C	SEARCH FOR PIVOT ELEMENT	TYPE0778
	40 AMAX=0.0	TYPE0779
	45 DO 105 J=1,N	TYPE0780
	50 IF (IPIVOT(J)-1) 60, 105, 60	TYPE0781
	60 DO 100 K=1,N	TYPE0782
	70 IF (IPIVOT(K)-1) 80, 100, 740	TYPE0783
	80 IF (ABS(A-AMAX)-ABS(A(J,K))) 85, 100, 110	TYPE0784
	85 IROW=J	TYPE0785
	90 ICOLUMN=K	TYPE0786
	95 AMAX=A(J,K)	TYPE0787
	100 CONTINUE	TYPE0788
	105 CONTINUE	TYPE0789
	110 IPIVOT(ICOLUMN)=IPIVOT(ICOLUMN)+1	TYPE0790
C	INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL	TYPE0791
	120 IF (IROW=ICOLUMN) 140, 20, 140	TYPE0792
	140 DETERM=-DETERM	TYPE0793
	50 DO 200 L=1,N	TYPE0794
	60 SWAP=A(IROW,L)	TYPE0795
	70 A(IROW,L)=A(ICOLUMN,L)	TYPE0796
	200 A(ICOLUMN,L)=SWAP	TYPE0797

205	IF(M) 260, 260, 21.	TYPE0798
21	DO 250 L=1, M	TYPE0799
220	SHAP=0(IROW,L)	TYPE0800
230	B(IROW,L)=B(ICOLUMN,L)	TYPE0801
250	B(ICOLUMN,L)=SHAP	TYPE0802
260	INDEX(1,1)=IROW	TYPE0803
270	INDEX(1,2)=ICOLUMN	TYPE0804
310	PIVOT(1)=A(ICOLUMN,ICOLUMN)	TYPE0805
320	DETERM=DETERM+PIVOT(1)	TYPE0806
C	DIVIDE PIVOT ROW BY PIVOT ELEMENT	TYPE0807
330	A(ICOLUMN,ICOLUMN)=1.0	TYPE0808
340	DO 350 L=1,N	TYPE0809
350	A(ICOLUMN,L)=A(ICOLUMN,L)/PIVOT(1)	TYPE0810
355	IF(M) 360, 360, 34.	TYPE0811
360	DO 370 L=1,M	TYPE0812
370	B(ICOLUMN,L)=B(ICOLUMN,L)/PIVOT(1)	TYPE0813
C	REDUCE NON-PIVOT ROWS	TYPE0814
380	DO 590 L1=1,N	TYPE0815
390	IF((1-ICOLUMN) 400, 550, 4.)	TYPE0816
400	T=A(L1,ICOLUMN)	TYPE0817
420	A(L1,ICOLUMN)=.0	TYPE0818
430	DO 450 L=1,N	TYPE0819
450	A(L1,L)=A(L1,L)-A(ICOLUMN,L)*T	TYPE0820
455	IF(M) 550, 450, 4A.	TYPE0821
460	DO 540 L=1,M	TYPE0822
500	B(L1,L)=B(L1,L)-B(ICOLUMN,L)*T	TYPE0823
550	CONTINUE	TYPE0824
C	INTERCHANGE COLUMNS	TYPE0825
600	DO 710 I=1,N	TYPE0826
610	LEN=1-1	TYPE0827
620	IF (INDEX(1,1)-INDEX(1,2)) 640, 710, 630	TYPE0828
630	JROW=INDEX(1,1)	TYPE0829
640	JCOLUMN=INDEX(1,2)	TYPE0830
650	DO 700 K=1,N	TYPE0831
660	SHAP=A(K,JROW)	TYPE0832
670	A(K,JROW)=A(K,JCOLUMN)	TYPE0833
700	A(K,JCOLUMN)=SHAP	TYPE0834
705	CONTINUE	TYPE0835
710	CONTINUE	TYPE0836
760	RETURN	TYPE0837
	END	TYPE0838
	END(1,1,1,1,1)	TYPE0839